Carnegie Mellon
Software Engineering Institute

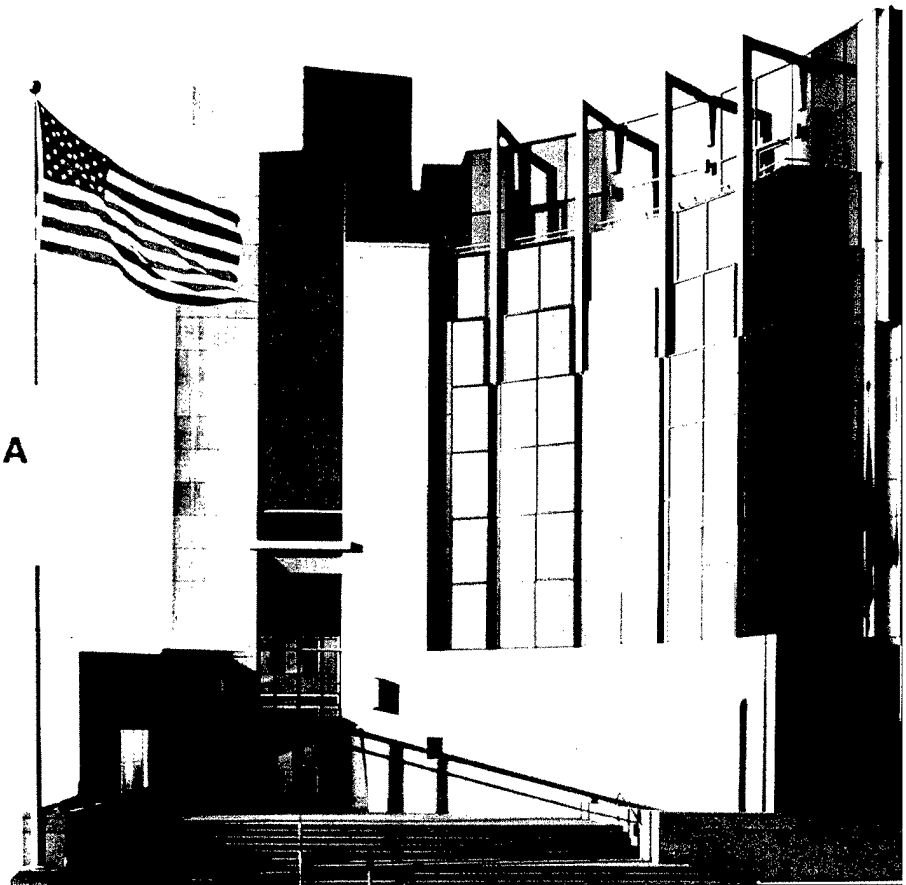
Software Process Improvement and Product Line Practice: Building on Your Process Improvement Infrastructure

Lawrence G. Jones

September 2004

Product Line Practice Initiative

Technical Note
CMU/SEI-2004-TN-044



Software Process Improvement and Product Line Practice: Building on Your Process Improvement Infrastructure

Lawrence G. Jones

September 2004

Product Line Practice Initiative

Technical Note
CMU/SEI-2004-TN-044

This work is sponsored by the U.S. Department of Defense.

The Software Engineering Institute is a federally funded research and development center sponsored by the U.S. Department of Defense.

This work was created in the performance of Federal Government Contract Number F19628-00-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center. The Government of the United States has a royalty-free government-purpose license to use, duplicate, or disclose the work, in whole or in part and in any manner, and to have or permit others to do so, for government purposes pursuant to the copyright license under the clause at 252.227-7013.

For information about purchasing paper copies of SEI reports, please visit the publications portion of our Web site (http://www.sei.cmu.edu/publications/pubweb.html).

Contents

Abstract

Many organizations with an existing process improvement initiative are also considering a software product line adoption initiative. Managers in these organizations often ask how they can build on their process improvement work and reconcile these two significant change initiatives. This technical note addresses one aspect of this question: how a process improvement infrastructure can provide a foundation for product line adoption.

1 Introduction

Today software process improvement (SPI) is a widely accepted practice. Articles on SPI appear regularly in technical and trade journals [McConnell 02], and impressive return on investment (ROI) figures are routinely reported [Ferguson 99, Goldenson 95, Zahran 97]. Additionally, the Software Engineering Coordinating Committee, a joint committee of the IEEE Computer Society and the Association for Computing Machinery (ACM) [IEEE CS 04b], has identified software process and related topics as foundational knowledge areas in both the Software Engineering Body of Knowledge [IEEE 04] and the Software Engineering Education Knowledge [IEEE CS 04a]. Practitioner acceptance is evidenced by the large annual Software Engineering Process Group (SEPG) conferences in the United States, Latin America, Europe, and Asia. Furthermore, there are currently about 130 Software Process Improvement Network (SPIN) chapters worldwide with others in the wings promising to bring the number to 150.

Software product line practice is a newer technology and appears to be in a position similar to where SPI was about a decade ago. Motivating product line technology is the increasing realization among organizations that they can no longer afford to develop multiple software products one product at a time. They are pressured to introduce new products and add functionality to existing ones at a rapid pace. They have explicit needs to achieve large-scale productivity gains, improve time to market, maintain a market presence, compensate for an inability to hire, leverage existing resources, and achieve mass customization. Many organizations are finding that the practice of building sets of related systems together can yield remarkable quantitative improvements in productivity, time to market, product quality, and customer satisfaction. These organizations are adopting a product line approach for their software systems.

Particularly exciting has been evidence of the increased benefits achieved when a product line approach is coupled with SPI. John Vu of the Boeing Company has studied the improvements in organizations with highly mature processes [Vu 00]. His studies show that the benefits of applying SPI in a single-product context tend to level off at the higher maturity levels. However, when this improvement includes a shift to a product line approach, the productivity increase is significant, as much as 70%. Similarly, Cummins Engine Inc. estimates that process improvement alone resulted in a benefit-to-cost ratio of between 2:1 and 3:1, while software product line practice, applied in addition to software process discipline, resulted in a benefit-to-cost ratio of 10:1 [Clements 02].

Thus, many organizations with a process improvement program in place are now looking at adopting a software product line approach. In particular, many organizations have successfully based their software engineering process efforts on the Capability Maturity

Model® (CMM®) for Software [Paulk 95] or its descendants, the Capability Maturity Model Integration℠ (CMMI℠) models [SEI 04a].[1] Some of these organizations are also using the SEI *Framework for Software Product Line Practice℠* (henceforth referred to as the Framework) as a model for product line practice [Clements 04].[2] One of the first things such organizations notice is that process improvement and software product line practice have tantalizing similarities. However, as they delve into implementation details, they find enough differences to be confused. The purpose of this technical note is to clarify how to exploit an existing process improvement effort to jumpstart software product line adoption.

Organizational initiatives like process improvement and product line adoption are all about change. Successful change addresses at least two dimensions: the technical aspects of the change and the organizational or "people" aspects of the change. Jones and Soule address an important aspect of the technical dimension by showing key relationships between the CMMI models and the Framework [Jones 02]. The gist of the comparison is that while CMMI process areas may provide a basis for some corresponding product line practice areas, there are always special product line "twists" that go beyond the CMMI. The "people" dimension of successful change is often handled by a supporting improvement infrastructure. This technical note will address how to use an existing process improvement infrastructure to support software product line adoption.

While we will make particular reference to the CMMI models, the general ideas are independent of the model for process discipline.[3] Also, while this technical note refers to many process improvement practices, it is not a tutorial on such practices. They are addressed frequently at the SEPG conferences [SEI 04b]; also see Zahran's work for more information [Zahran 97].

In Section 2, we discuss specific aspects of the infrastructure and how they can support software product line adoption. We conclude in Section 3 with a brief summary.

® Capability Maturity Model and CMM are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University.

℠ CMMI, CMM Integration, and Framework for Software Product Line Practice are service marks of Carnegie Mellon University.

1 Because CMMI models go beyond software processes, we will hereafter use the more general term *process improvement*.

2 For an overview of CMMI models and the Framework, see the work of Jones and Soule [Jones 02].

3 Other models besides CMMI can provide process discipline for appropriately supporting a product line approach. See the work of Zahran for several examples including ISO 15504, ISO 9001, and BOOTSTRAP [Zahran 97]. While the details will differ, the general concepts in Jones and Soule's work are relevant when comparing other process models to the Framework [Jones 02].

2 Leveraging Process Improvement Infrastructure to Support Product Line Practice

An established process improvement infrastructure typically includes at least the following elements:

- oversight and implementation
- process assets
- a training infrastructure
- other change management assets

In this section, we explain each element in the process improvement context and then explore how it can be augmented (or emulated) to support software product line practice.

2.1 Oversight and Implementation

The typical organizational roles and elements that oversee and implement process improvement are the sponsor, the management steering group (MSG), the process group (PG), and process action teams (PATs). We explain each of these and their applicability in a product line approach below.

The sponsor role provides executive support for process improvement activities. This support includes balancing tradeoffs, establishing priorities, allocating resources, and providing executive leadership. A particular organization might have a chain of sponsorship. Successful product line adoption requires the same type of sponsorship, and the same approach for sponsoring process improvement can be used to sponsor product line adoption. If these two initiatives have different sponsors, it is essential that their sponsorship be coordinated. If the push for adopting software product lines did not originate with the appropriate executives, the sponsorship and advocacy-building tactics that are among typical change-management assets are useful here (see Section 2.4).

The MSG oversees the direction and progress of an organization's process improvement effort, primarily by managing the PG. Typically, a strategic process improvement plan is used to guide the effort with the MSG as the owner of the plan and the PG as the implementation agent. The MSG consists of key managers with a stake in the organization's processes. The MSG structure for process improvement is a useful model to copy for software product line

practice. Thus, a product line steering group (PLSG), owning and following a product line adoption plan, could

- support and direct the software product line manager and his/her staff (e.g., a product line group as described below)
- set direction for the product line and arbitrate conflicting needs
- provide general support for the product line including advocacy and reinforcement of sponsorship through the organizational chain
- coordinate with the MSG

The overlap in the membership of the MSG and PLSG might be significant or even complete. In any case, there should be well-defined charters as well as roles and responsibilities specific to the needs of the two initiatives. These initiatives should be managed like any well-managed project and should not be treated as just another generic management task.

The PG, as directed by the MSG, is the group that facilitates the definition, installation, maintenance, and improvement of an organization's process assets according to a strategic process improvement plan. The PG provides continuity, coordination, and technical support for the PATs. The PG's structure, roles, procedures, and other assets can provide a good model for a comparable product line group (PLG). Following the PG model, the PLG would be the implementing agent for the product line adoption plan. While the PLG can benefit from process-oriented assets of the PG, many tasks facilitated by the PLG are not process oriented such as building a business case for the product line, defining the product line scope, developing a funding approach, and doing a market analysis for product line potential. For these tasks, the PLG would have to chart most of its own way. It would be natural for the software product line manager to manage the PLG as directed by the PLSG. Since the PG and PLG will be introducing significant organizational change, close coordination is necessary at the working level.

The PATs are ad hoc teams that implement specific portions of the process improvement plan (e.g., the definition and rollout of a particular process). The PG serves as a resource for the PATs, and PG members often lead or at least participate in various PATs. Software product line practice will affect many organizational processes. Thus, PATs creating or adapting processes to support a product line should include team members who can represent product line interests. For the non-process-related aspects (see some of the PLG tasks previously noted, such as defining the product line scope), the structure, procedures, and other assets for PATs (maintained by the PG) should prove useful for product line purposes. Product line action teams could be constituted as necessary to carry out these product-line-specific tasks.

2.2 Process Assets

Common, useable process assets are essential to process standardization. The CMMI Organizational Process Definition process area describes the practices[4] for establishing and maintaining an organization's set of process assets [CMMI PT 02, pgs. 331-347]. These assets include standard processes, life-cycle models, tailoring criteria and guidelines, a measurement repository, and a process asset library. While full implementation of these practices is required at maturity level 3 in the staged model representation, many organizations start building their process assets early in the process improvement initiative.

The CMMI models note that there are many ways to define the repositories for process assets. For the purposes of this discussion, we assume that the process asset library is the overall repository used to store and make available all the potentially useful process assets. The CMMI models provide examples of the types of artifacts that might go into a process asset library including

- policies
- process descriptions
- procedures
- plans (e.g., development, quality assurance, testing, piloting, and rollout)
- process aids (e.g., standards, checklists, templates, documents, and document fragments)
- lesson learned reports

Because product line practice requires significant commonality of approach within an organization, augmenting the process asset library can be an important task for supporting product line practice. Product line assets that are process oriented would likely be included in the library as a matter of course. Usability and accessibility considerations for other product line assets (e.g., the business case and guidelines for its creation and maintenance) should influence where and how such assets are stored.

2.3 Training Infrastructure

Training is an integral part of any technology change and is crucial for helping the change to be lasting. CMMI models address training in two ways. First, they treat training as a recurring generic practice (GP). GPs support the institutionalization of *all* process areas to ensure that the processes associated with the process area will be effective, repeatable, and lasting. Second, CMMI models cover training in a separate process area, Organizational Training. An organization that has implemented the CMMI Organizational Training process

[4] These practices are summarized in the appendix.

area[5] has an excellent infrastructure for supporting software product line practice. This infrastructure includes processes to

- determine training needs
- determine the level of responsibility for training
- plan and deliver training

A training organization is often responsible for managing the training program. Clearly, this training resource can be applied to product-line-specific needs and provides a solid basis for the "Training" practice area of the Framework.

2.4 Other Change Management Assets

Successful process improvement involves developing change management skills and tools that don't necessarily have a process focus but provide useful underpinnings. These assets are often developed and maintained by the process group. Such assets have obvious applicability in supporting product line adoption. Examples include

- resistance management skills and tools
  - These skills and tools include assets to analyze change resistance within an organization and a capability to plan and execute strategies to overcome resistance. Example assets include resistance-focused organizational survey tools, resistance management models, common resistance management strategies, resistance management training, and plans to address resistance.[6]
- sponsorship and advocacy development and nurturing
  - The ability to be a good sponsor and advocate for organizational direction is an important leadership skill. Beyond the broad literature on leadership, Senge provides a model and techniques for leading a "learning organization" that embraces positive change [Senge 90, Senge 94]. Deimel, Maher, and Myers provide succinct practical guidance on sponsor building in the Managing Technological Change course.[6] Sponsorship and advocacy-building techniques for process improvement are directly applicable for product lines.
- communications strategies
  - Communications throughout the organization are a critical success factor for change. Communications approaches for process improvement are also useful for product lines.
- team creation and performance building

[5] The specific goals and specific practices of this process area are summarized in the appendix.

[6] One valuable tool for this type of training is the SEI's Managing Technological Change course. For more information, go to http://www.sei.cmu.edu/products/courses/mtc.html.

- Successful change is a team effort throughout the organization. Team-building techniques useful for process improvement are equally applicable for software product lines. Scholtes' work is one example of a guidebook of techniques for building and growing effective teams [Scholtes 88].

3 Summary

Process discipline is an essential foundation for software product line practice. However, success in software product lines requires mastery of many other essential practice areas. In particular, software product line practice requires attention to the software *product* as well as the *process*. Nevertheless, organizations with a solid process improvement infrastructure have a significant basis for supporting product line adoption. This technical note has provided several ideas for how to exploit an existing process improvement infrastructure in order to adopt a product line approach more quickly and cheaply.

Appendix A Selected CMMI Goals and Practices

This technical note makes particular use of the concepts of two CMMI model process areas: Organizational Process Definition and Organizational Training. This appendix lists the goals and practices associated with those model components and uses the following abbreviations: specific goal (SG) and specific practice (SP).

Organizational Process Definition

The purpose of Organizational Process Definition is to establish and maintain a useable set of organizational process assets.

SG 1 Establish Organizational Process Assets

- SP 1.1 Establish Standard Processes
- SP 1.2 Establish Life-Cycle Model Descriptions
- SP 1.3 Establish Tailoring Criteria and Guidelines
- SP 1.4 Establish the Organization's Measurement Repository
- SP 1.5 Establish the Organization's Process Asset Library

Organizational Training

The purpose of Organizational Training is to develop the skills and knowledge of people so they can perform their roles effectively and efficiently.

SG 1 Establish an Organizational Training Capability

- SP 1.1 Establish the Strategic Training Needs
- SP 1.2 Determine Which Training Needs Are the Responsibility of the Organization
- SP 1.3 Establish an Organizational Training Tactical Plan
- SP 1.4 Establish Training Capability

SG 2 Provide Necessary Training

- SP 2.1 Deliver Training
- SP 2.2 Establish Training Records
- SP 2.3 Assess Training Effectiveness

Bibliography

URLs are valid as of the publication date of this document.

[Clements 02] Clements, P. & Northrop, L. *Software Product Lines: Practices and Patterns*. Reading, MA: Addison-Wesley, 2002.

[Clements 04] Clements, P. & Northrop, L. *A Framework for Software Product Line Practice, Version 4.2*. http://www.sei.cmu.edu/plp/framework.html (2004).

[CMMI PT 02] CMMI Product Team. *CMMI for Systems Engineering/Software Engineering/Integrated Product and Process Development/Supplier Sourcing, Version 1.1, Staged Representation (CMMI-SE/SW/IPPD/SS, V1.1, Staged)* (CMU/SEI-2002-TR-012, ADA339907). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2002. http://www.sei.cmu.edu /publications/documents/02.reports/02tr012.html

[Ferguson 99] Ferguson, P. et. al. *Software Process Improvement Works!* (CMU/SEI-99-TR-027, ADA371804). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1999. http://www.sei.cmu.edu/publications/documents/99.reports/99tr027 /99tr027abstract.html

[Goldenson 95] Goldenson, D. & Herbsleb, J. *After the Appraisal: A Systematic Survey of Process Improvement, Its Benefits, and Factors that Influence Success* (CMU/SEI-95-TR-009, ADA302225). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 1995. http://www.sei.cmu.edu/publications/documents /95.reports/95.tr.009.html

[IEEE 04] Institute of Electrical and Electronics Engineers, Inc. *Guide to Software Engineering Body of Knowledge*. http://www.swebok.org/ (2004).

[IEEE CS 04a] IEEE Computer Society. *Computing Curricula Software Engineering (CCSE)*. http://sites.computer.org/ccse/ (2004).

[IEEE CS 04b] IEEE Computer Society. *Software Engineering Coordinating Committee (SWECC).* http://www.computer.org/tab/swecc.htm (2004).

[Jones 02] Jones, L. & Soule, A. *Software Process Improvement and Product Line Practice: CMMI and the Framework for Software Product Line Practice* (CMU/SEI-2002-TN-012, ADA403868). Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2002. http://www.sei.cmu.edu/publications/documents/02.reports/02tn012.html

[McConnell 02] McConnell, Steve. "The Business of Software Improvement." *IEEE Software 19*, 4 (July/August 2002): 5-7.

[Paulk 95] Paulk, M. et al. *The Capability Maturity Model: Guidelines for Improving the Software Process.* Reading, MA: Addison-Wesley, 1995.

[Scholtes 88] Scholtes, P. *The Team Handbook.* Madison, WI: Joiner Associates Inc., 1988.

[SEI 04a] Software Engineering Institute. *CMMI Models and Modules.* http://www.sei.cmu.edu/cmmi/models/#models (2004).

[SEI 04b] Software Engineering Institute. *SEPG 2005.* http://www.sei.cmu.edu/sepg/ (2004).

[Senge 90] Senge, P. *The Fifth Discipline: The Art and Practice of the Learning Organization.* New York, NY: Doubleday, 1990.

[Senge 94] Senge, P. *The Fifth Discipline Fieldbook: Strategies and Tools for Building a Learning Organization.* New York, NY: Doubleday, 1994.

[Vu 00] Vu, J. "Findings of the Managing Software Innovation and Technology Change Workshop." *Proceedings of SEPG 2000* (CD-ROM). Seattle, WA, March 20-23, 2000. Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2000.

[Zahran 97] Zahran, S. *Software Process Improvement: Practical Guidelines for Business Success.* Reading, MA: Addison-Wesley, 1997.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
(Leave Blank)	September 2004	Final

4. TITLE AND SUBTITLE	5. FUNDING NUMBERS
Software Process Improvement and Product Line Practice: Building on Your Process Improvement Infrastructure	F19628-00-C-0003

6. AUTHOR(S)

Lawrence G. Jones

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
Software Engineering Institute Carnegie Mellon University Pittsburgh, PA 15213	CMU/SEI-2004-TN-044

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER
HQ ESC/XPK 5 Eglin Street Hanscom AFB, MA 01731-2116	

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT	12B DISTRIBUTION CODE
Unclassified/Unlimited, DTIC, NTIS	

13. ABSTRACT (MAXIMUM 200 WORDS)

Many organizations with an existing process improvement initiative are also considering a software product line adoption initiative. Managers in these organizations often ask how they can build upon their process improvement work and reconcile these two significant change initiatives. This technical note addresses on aspect of this question: how a process improvement infrastructure can provide a foundation for product line adoption.

14. SUBJECT TERMS	15. NUMBER OF PAGES
CMMI, software product line practice, software process improvement, software product line adoption	18

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	UL